Journal Article

Behavioral Study of Software-Defined Network Parameters Using Exploratory Data Analysis and Regression-Based Sensitivity Analysis

Akinsolu, M. O., Sangodoyin, A. O. and Uyoata, E. U.

mathematics

MDPI

Article

Behavioral Study of Software-Defined Network Parameters Using Exploratory Data Analysis and Regression-Based Sensitivity Analysis

Mobayode O. Akinsolu [1,*], Abimbola O. Sangodoyin [2,*] and Uyoata E. Uyoata [3,*]

[1] Faculty of Arts, Science and Technology, Wrexham Glyndŵr University, Wrexham LL11 2AW, UK
[2] School of Mathematics and Computer Science, University of Wolverhampton, Wolverhampton WV1 1LY, UK
[3] Department of Electrical and Electronics Engineering, Modibbo Adama University, Yola P.M.B. 2076, Nigeria
* Correspondence: m.o.akinsolu@ieee.org (M.O.A.); asangodoyin@ieee.org (A.O.S.); uyoataue@mautech.edu.ng (U.E.U.)

Abstract: To provide a low-cost methodical way for inference-driven insight into the assessment of SDN operations, a behavioral study of key network parameters that predicate the proper functioning and performance of software-defined networks (SDNs) is presented to characterize their alterations or variations, given various emulated SDN scenarios. It is standard practice to use simulation environments to investigate the performance characteristics of SDNs, quantitatively and qualitatively; hence, the use of emulated scenarios to typify the investigated SDN in this paper. The key parameters studied analytically are the jitter, response time and throughput of the SDN. These network parameters provide the most vital metrics in SDN operations according to literature, and they have been behaviorally studied in the following popular SDN states: normal operating condition without any incidents on the SDN, hypertext transfer protocol (HTTP) flooding, transmission control protocol (TCP) flooding, and user datagram protocol (UDP) flooding, when the SDN is subjected to a distributed denial-of-service (DDoS) attack. The behavioral study is implemented primarily via univariate and multivariate exploratory data analysis (EDA) to characterize and visualize the variations of the SDN parameters for each of the emulated scenarios, and linear regression-based analysis to draw inferences on the sensitivity of the SDN parameters to the emulated scenarios. Experimental results indicate that the SDN performance metrics (i.e., jitter, latency and throughput) vary as the SDN scenario changes given a DDoS attack on the SDN, and they are all sensitive to the respective attack scenarios with some level of interactions between them.

Keywords: exploratory data analysis; linear regression; sensitivity analysis; software-defined networks

MSC: 68U01

Citation: Akinsolu, M.O.; Sangodoyin, A.O.; Uyoata, U.E. Behavioral Study of Software-Defined Network Parameters Using Exploratory Data Analysis and Regression-Based Sensitivity Analysis. *Mathematics* **2022**, *10*, 2536. https://doi.org/10.3390/math10142536

Academic Editor: Daniel-Ioan Curiac

Received: 31 May 2022
Accepted: 15 July 2022
Published: 21 July 2022

1. Introduction

Data networks are becoming increasingly more ubiquitous and pervasive owing to a plethora of applications that rely heavily on data-driven paradigms and robust information exchange. Amongst these applications, the Internet of Things (IoT) [1] and its apparent subset, the Industrial Internet of Things (IIoT) [2], appear to be the most dominant [3]. This is primarily because IoT and IIoT are continuously creating and offering innovative solutions that are based on very robust, and sometimes unprecedented levels of interconnectivity between devices, systems and networks [1–3]. Despite added advantages such as the decentralization of control on industrial shop floors and ubiquity of process management [4], enhanced visualization of operations for cognition and informed decision-making [5], and higher levels of inter-connectivity for latency critical operations [6,7], present-day IoT and IIoT data network architectures that still suffer from the drawback of complexity of management as they grow and become more subjected to uneven dynamic

behavioural changes [2,3]. Software-defined networking is an approach aimed at addressing the complexity of management in contemporary data networks such as IoT and IIoT data networks [2,8].

Generally, software-defined networks (SDNs) are implemented to mainly decouple control management and data forwarding in present-day data networks [8]. To affect this, SDNs support the flexibility, programmability, and openness of many data networks today [8]. In a typical SDN, a network operating system, called the SDN controller, holds the entire data network information (including topology, dynamic changes, application requirements and security requirements) and network administrators or operators are able to dynamically configure or program routines or instructions implemented on network forwarding devices to guarantee optimal allocation and utilization of network resources. With this network architecture, forwarding equipment in SDNs adopt a unified interface to exchange data and/or information, usually in the form of packets, with the SDN controller, which in turn can obtain the operational status of the data network for efficient traffic engineering and other network services [8]. According to available literature, the primary challenge with the higher programmability, flexibility and openness introduced into modern data networks such as IoT and IIoT networks by SDN-based architectures is the increased susceptibility to security breaches such as distributed denial of service (DDoS) flooding attacks [9].

By convention, DDoS flooding attacks are well-organized attacks that emanate from several compromised hosts that target the nodes or end-users' devices on the network with the goal of usurping the available network bandwidth or rendering the nodes and end-user devices completely unavailable. Last year, about 20 DDoS flooding attacks were launched every minute globally [10], and they are often classified as the most dangerous malicious traffic on the internet [11]. Consequently, DDoS flooding attacks have been extensively discussed in available literature [12–15]. Many researchers in this domain agree that there is not a unified way for the launching of DDoS flooding attacks; typically, perpetrators clandestinely engage a botnet (entire networks of infected devices) to launch attacks [12,14]. As a result, operators of the device nodes on attacked networks are often blindsided and unaware of the fact that attacks are emanating from their devices and internet protocol (IP) addresses.

Even though DDoS flooding attacks on SDNs have been discussed extensively in the literature [12–15], a behavioral study of the variants of DDoS flooding attacks that can be launched on present-day SDNs using the performance metrics of the data network is still a relatively grey area, as discussed further in Section 2. Several research efforts have evaluated the efficiency and performance of SDNs using throughput, jitter and response time metrics [15–19]. Thus, they are arguably the most popular SDN performance metrics. A behavioral study of these popular SDN performance metrics under popular SDN DDoS states will aid network administrators in characterizing and distinguishing DDoS flooding attacks on SDN platforms. Consequently, more robust mitigation techniques, response plans and actions can be developed to ascertain the reliability and availability of present-day and future SDNs. In this work, a behavioral study of SDN performance metrics is carried out for various scenarios of DDoS flooding attacks using the jitter, latency and throughput of an emulated SDN [15,16].

The emulated SDN model used, and the associated dataset explored and analyzed, have been typified in literature to be representative of real-world SDN and SDN scenarios [15,16] (see Section 3). Hence, the further analysis carried out in this work using the SDN model and its data are an additional work to the outcomes detailed in [15,16]. Particularly, the performance metrics of the emulated SDN subjected to hypertext transfer protocol (HTTP) flooding, transmission control protocol (TCP) flooding, and user datagram protocol (UDP) flooding, given a distributed denial-of-service (DDoS) attack are analyzed to make the following main contributions:

- Behavioural study of popular SDN performance metrics (i.e., jitter, latency and throughput) under real-world SDN operations (normal SDN operations and when the SDN is subjected to popular variants of DDoS flooding attacks);
- Regression-based sensitivity analysis (RSA) of popular SDN performance metrics (i.e., jitter, response time and throughput) to ascertain the pairwise interactions between them for real-world SDN operations (normal SDN operations and when the SDN is subjected to popular variants of DDoS flooding attacks);
- Low-cost inference-driven assessment and characterization of real-world SDN operations.

The remaining part of this paper is organized as follows: Section 2 summarily discusses the related work to reemphasize the practical need of the work carried out, Section 3 describes the SDN configuration and network setup, Section 4 presents the step-by-step procedures required in the proposed approach, Section 5 details the behavioral study and sensitivity analysis of the SDN performance metrics for the popular variants of DDoS flooding attacks, Section 6 highlights the guidelines for inference-based characterization of DDoS flooding attacks on SDNs, and the concluding remarks are provided in Section 7.

2. Related Work

In comparison to traditional networks, the effect of DDoS attacks can be severe in SDNs when attacks are launched against the infrastructure, control, and application layers (see Section 3). Hence, a behavioral study of the performance metrics that predicate the proper functioning of SDNs is highly essential for the instantaneous or on-the-fly detection and evaluation of abnormal SDN operations. Typically, network performance metrics are studied to find possible ways of enhancing the quality of service (QoS) in SDN environments. For example, in [19], the response time and throughput metrics of the SDN have been utilized to predict packet scheduler activities for real-time online interactive applications (ROIA), hierarchical token bucket (HTB), stochastic fairness queueing (SFQ), and random early detection (RED). As a result, the HTB packet scheduler and SFQ packet scheduler were deduced to be better in terms of response time and throughput, respectively, in comparison to the other common architectures investigated. Even though this work correctly demonstrates that a representative subset of the SDN flows can be monitored to effect QoS monitoring, an extensive behavioral study of the SDN metrics considered has not been conducted.

In [17], the jitter metric and two other metrics (end-to-end delay and throughput linked to packet loss and link utilization) have been studied to improve the QoS for SDN-based robotic cyber-physical systems. The primary goal of the methodology presented in [17] is to monitor network links and react to abnormal network states by dynamically migrating flows to more stable alternative routes. As a result, it was shown that, as the number of flows and emulation time varies, the SDN metrics (i.e., jitter, end-to-end delay, and throughput) are altered. In each case, the QoS-aware routing scheme (QRS) proposed in [17] opted for a path or route with less jitter, lower end-to-end delay (indicating lower packet loss and higher available bandwidth), and more throughput. However, exploratory analysis and statistical or inferential evaluation of the alterations of the SDN metrics in the various SDN states were not carried out in [17]. To better validate the robustness of SDN architectures in comparison to traditional network architectures (in the context of future generation networks that are desired to be flexible, scalable, and highly secure), Mininet and Graphical Network Simulator-3 (GNS3) have been used to emulate and typify an SDN and a conventional network, respectively, for the evaluation of their respective metrics (latency and jitter) in [18]. Even though the experimental study of the latency (SDN performance metric in [18]) revealed that the SDN offers an average latency over three times lower than the traditional network, a behavioral study of the network performance metrics was neither presented nor discussed in [18].

A relatively in-depth analysis of SDN parameters or performance metrics, local sensitivity analysis of throughput, response time, and jitter metrics under various SDN states

has been reported in [16] by the authors. The results in [16] show that the throughput, jitter, and response time metrics of the SDN are all statistically sensitive to the changes in the SDN states from normal operations to abnormal operations, following popular DDoS flooding attacks on the SDN. The sensitivities deduced in [16] are relative to the change of a single SDN parameter or metric value at a time, i.e., throughput or response time or jitter, with the jitter metric being the most sensitive. Hence, a global analysis methodology that examines the sensitivities of the SDN parameters or metrics relative to their entire distributions or trends presents a research gap. Sequel to the work in [16] and as a way of verifying that the states of the SDN (categorical or discrete variables) can be effectively mapped to the measures of its key performance metrics (numerical or continuous variables), four popular classifiers (machine learning algorithms) are investigated in [15] for the predictive modeling of the SDN states (normal state and DDoS flooding attack states), given a set of SDN metrics (i.e., throughput, jitter, and response time). All the classifiers investigated in [15] show good efficiency in detecting and classifying the SDN state. However, a behavioral study of the SDN parameters has not been carried out per se in [15].

Based on the works discussed above, it can be said that most of the recent works in the available literature tend to proffer solutions for enhancing QoS, detecting and classifying the state of the SDN, given an attack such as a DDoS flooding attack. Therefore, exploratory data analysis and regression-based sensitivity analysis to characterize and draw inferences on the behavior of the SDN using key performance metrics, in particular, as carried out in this work and discussed in the subsequent sections, offers a new paradigm and insight into how to secure the SDN when it is subjected to abnormal state changes or transitions due to attacks such as DDoS flooding attacks, based on a more robust inferential assessment of the SDN.

3. SDN Architecture and Experimental Setup

The architecture of the SDN is explained summarily in this section to provide a background to the experimental setup that is explained in more details in this section as well.

3.1. SDN Architecture

Typically, SDNs are built on the separation of the data plane and the control plane of networks to allow the logical control of network devices [20]. As illustrated in Figure 1, controllers and forwarding devices constitute the basic architecture of the SDN. The controller sits as the decision-making entity and uses information from the applications running on devices in the network to make decisions for the data plane, whereas the forwarding devices engage in the actual message forwarding based on stated forwarding policies [21]. Forwarding devices could be implemented in hardware or software. For the work in this paper, the explained SDN architecture is used.

3.2. Experimental Setup

To set up the system model for the experiments in this work, a custom fat-tree topology having three layers of switches administered by a controller, and serving a group of end devices or hosts was created. A pictorial representation is given in Figure 2. The network design was carried out in an open-source network emulator, specifically Mininet. Mininet emulates realistic virtual networks consisting of links, switches, and end devices, and it runs a real Linux kernel. Note that it is not unconventional to use emulators or simulation techniques for network management to observe the behavior and characteristics of the network system before deployment [22–24]. Mininet is a very popular emulator in the SDN research domain, and networks modeled in the Mininet environment have been reported in several works to typify real-world SDN scenarios [22–24]. Hence, its choice for the typification of the SDN in this work.

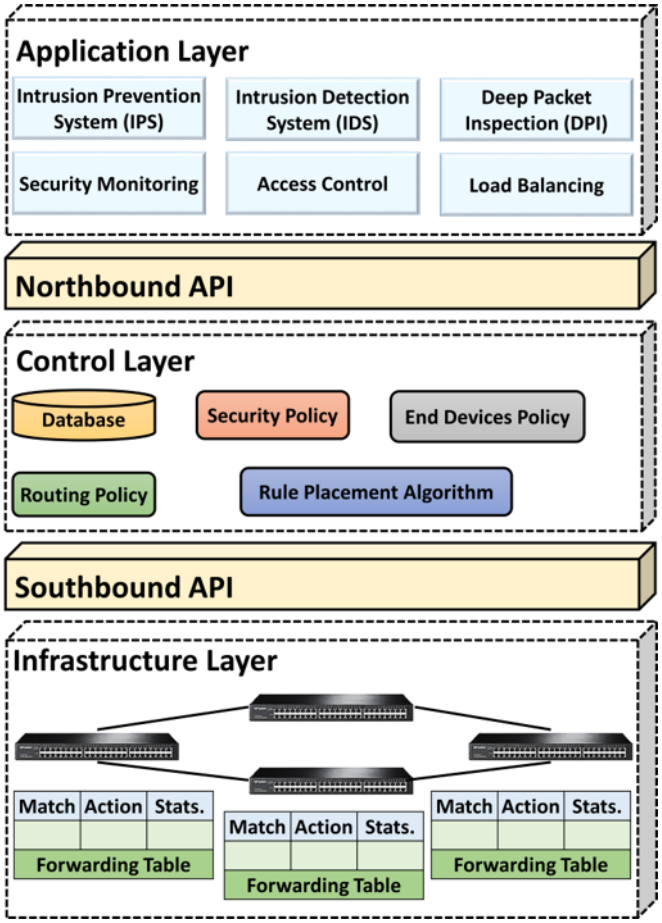

Figure 1. SDN architecture.

As shown in Figure 2, the custom topology is implemented on a 32 GB RAM Xeon processor with Kali Linux as the base operating system. The floodlight controller is deployed in VirtualBox running Ubuntu 18.10 LTS and Mininet emulator is deployed on Ubuntu 16.10 LTS. Sixteen interconnected hosts or end nodes and 10 OpenFlow switches were considered for the experimentation. The links between the hosts and the switches support throughput up to 100 Mbps. Although experimental, this size of the network is easily applicable to enterprise cases and campus network scenarios. To create non-malicious traffic between the network nodes, "iperf" and "ping" commands were used. Low Orbit Ion Cannon (LOIC) was used to launch DDoS attacks on the network server from designated compromised six hosts (hosts six–eight and hosts 10–12). These attacks (HTTP, TCP, and UDP DDoS flooding attacks) were launched for 15 min each, i.e., for each round of the experiments, subjecting the network server to 45 min of flooding attacks in total. During the duration of these attacks, generated system properties were recorded for analysis.

For the work in this paper, the system properties of interest recorded and examined are throughput, jitter, and response time. Their generated values were stored and used to analyze the network performance. Conventionally, throughput is the actual amount of traffic flowing within the network per time, jitter is the variation in the time delay between the transmission and successful delivery of packets within the network connection, and response time is the elapsed time between the successful initiation and successful termination of a task within the network. More contextual relevance of these system properties is provided in Section 3.3. After the initial custom network setup using Mininet, SDN data were generated using the following steps:

- **Step 1:** Confirm connectivity between hosts on the network using "pingall" command.
- **Step 2:** Using "iperf", create UDP and TCP servers to listen on different ports of the network.

- **Step 3:** Use hosts one and two to send ping requests to TCP, UDP, and HTTP servers and measure throughput, jitter, and response time.
- **Step 4:** Obtain the throughput, jitter, and response time for 15 min without attack (normal state) from the listening server port.
- **Step 5:** Use LOIC on compromised hosts within the system to launch TCP DDoS flooding attack for 15 min and obtain the throughput, jitter, and response time.
- **Step 6:** Use LOIC on compromised hosts within the system to launch UDP DDoS flooding attack for 15 min and obtain the throughput, jitter, and response time.
- **Step 7:** Use LOIC on compromised hosts within the system to launch HTTP DDoS flooding attack for 15 min and obtain the throughput, jitter and response time.

Steps 5, 6 and 7 were implemented at different time intervals, and for each DDoS flooding attack launched, the port numbers were changed to avoid interference in each scenario. Each output (SDN data) obtained in the Linux environment was converted to a .txt file and Konstanz Information Miner (KNIME) was used to extract features of interest (i.e., throughput, jitter, and response time metrics).

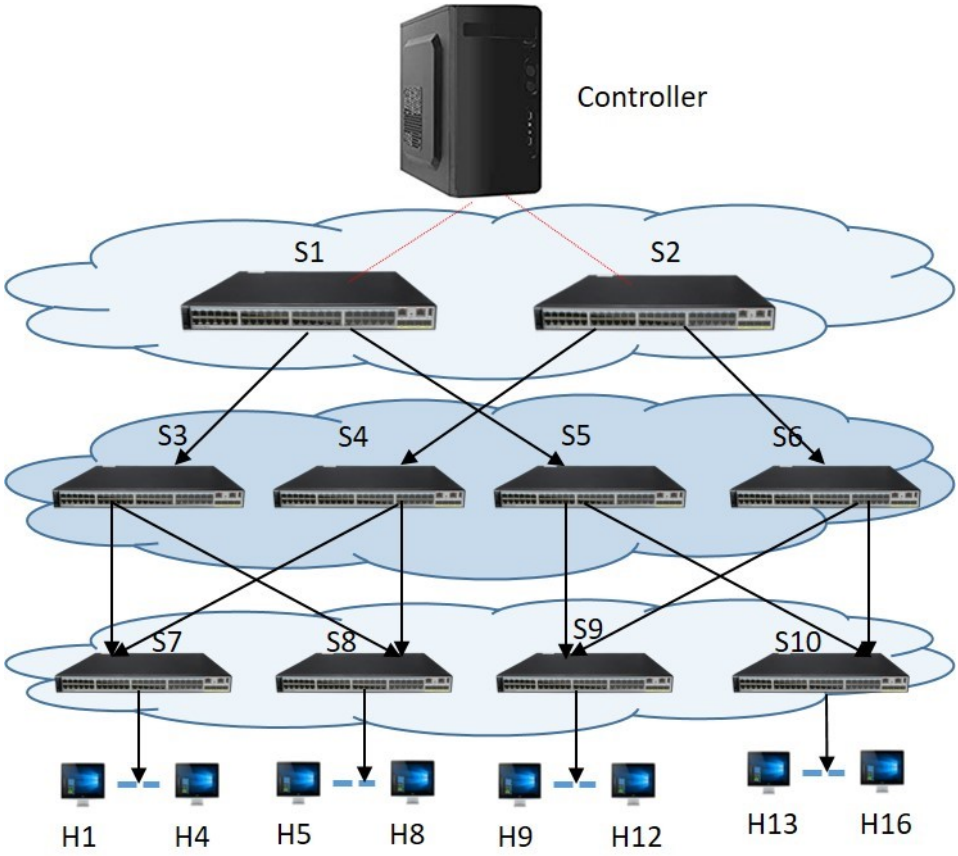

Figure 2. Modeled SDN tree topology.

3.3. SDN Performance Metrics

Network performance metrics help in predicting and preventing network downtime by identifying potential and unexpected errors on SDNs. Depending on the specific issues that affect SDNs, not every metric is going to be important to effectively measure network performance. There are some metrics that are essential for network administrators to consider as performance baselines. As discussed in Section 1, the network throughput (T_p), jitter (J_t) and response time (R_t) are core SDN performance metrics that allow for clear indication of the network state. Based on their definitions given in Section 3.2 and as later corroborated by their mean values in Tables 1–4, the deductions in Equation (1) can be made a priori and a posteriori. Note that there could be exceptions to Equation (1).

This will be particularly true for attacks that require the target node to respond by sending packets such as internet control message protocol (ICMP), as in the case of the UDP DDoS flooding attack [25]. In such a case, the T_p metrics can be high when the SDN is under attack. This is also corroborated in Table 3.

$$\text{Typical SDN scenarios} \begin{cases} T_p \rightarrow \text{High (SDN is not under attack)} \\ T_p \rightarrow \text{Low (SDN is under attack)} \\ R_t \rightarrow \text{Low (SDN is not under attack)} \\ R_t \rightarrow \text{High (SDN is under attack)} \\ J_t \rightarrow \text{Stable (SDN is not under attack)} \\ J_t \rightarrow \text{Vary a lot (SDN is under attack)} \end{cases} \tag{1}$$

In Equation (1), it can be said that a low value for the T_p metric and high values for the J_t and R_t metrics when the SDN is operational strongly suggest that the SDN is likely to be under attack, whereas a high value for the T_p metric and low values for the J_t and R_t metrics when the SDN is operational strongly suggest that the SDN is likely to not be under attack. In other words, the network parameter variations indicate that the SDN experiences changes due to attacks. Hence, their behavioral study is carried out in this work using the approach proposed in Section 4.

Table 1. Descriptive statistics for T_p, R_t and J_t (over 900 samples) for normal operating scenario.

Metric	Minimum	Maximum	Mean	Median	Standard Deviation
T_p	95.1000	95.9000	95.6332	95.6000	0.1402
R_t	0.0320	2.1200	0.2114	0.1980	0.1286
J_t	0.0040	0.4930	0.2271	0.1940	0.0943

Table 2. Descriptive statistics for T_p, R_t and J_t (over 900 samples) for the TCP DDoS flooding attack scenario.

Metric	Minimum	Maximum	Mean	Median	Standard Deviation
T_p	0.0000	95.9000	0.5441	0.0238	7.0999
R_t	0.2650	678.000	302.2676	299.000	110.6598
J_t	0.0040	0.4930	0.2271	0.1940	0.0943

Table 3. Descriptive statistics for T_p, R_t and J_t (over 900 samples) for the UDP DDoS flooding attack scenario.

Metric	Minimum	Maximum	Mean	Median	Standard Deviation
T_p	95.1000	95.9000	95.6332	95.6000	0.1402
R_t	0.1980	82.1000	25.3097	24.8000	7.3245
J_t	9.1610	18.4280	10.5100	10.1725	1.0496

Table 4. Descriptive statistics for T_p, R_t and J_t (over 900 samples) for the HTTP DDoS flooding attack scenario.

Metric	Minimum	Maximum	Mean	Median	Standard Deviation
T_p	0.0000	95.9000	0.7429	0.0000	8.3955
R_t	0.0200	1673.0000	49.1262	23.7000	90.9398
J_t	0.0040	0.4930	0.2271	0.1940	0.0943

4. Proposed Approach for the Behavioral Study

The flow diagram of the proposed approach adopted for the behavioral study is shown in Figure 3 and the essential steps are described summarily as follows:

- **Step 1:** Sampled SDN performance metrics (J_t, R_t and T_p) from the emulated SDN scenarios in Section 3 are declared as inputs into scripts and functions purpose-built for EDA and linear regression.
- **Step 2:** The EDA-based scripts and functions in Step 1 are firstly executed, and their outputs are analyzed to understand and visualize the trends or distributions of J_t, R_t and T_p metrics for inference-driven assessment and characterization of their alterations or variations.
- **Step 3:** Samples of J_t, R_t and T_p metrics from Step 1 are standardized at this stage, and their z-scores are used to derive an adaptive univariate response value for each J_t, R_t and T_p set or sample.
- **Step 4:** z-scores of the SDN metrics and the derived response values are used to build a linear regression model.
- **Step 5:** Coefficients and statistics of the linear regression model in Step 5 are used to account for the sensitivities and pairwise interaction effects between the J_t, R_t and T_p metrics for a more robust study of their behavior given the emulated SDN scenarios.

More details about the implementation of the proposed approach are presented in Section 5.

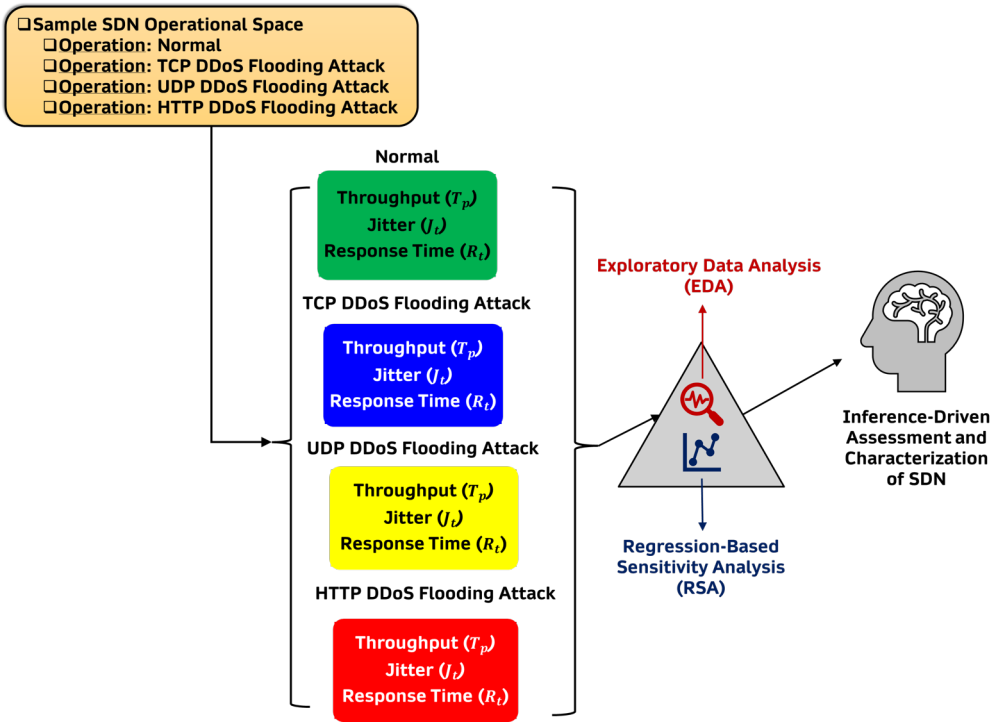

Figure 3. Flow diagram of the proposed behavioral study approach.

5. Analysis and Discussion

In this section, the SDN performance metrics generated based on the SDN configuration discussed, and the experiments conducted in Section 3 are visualized and critically investigated via univariate and multivariate, graphical, and non-graphical EDA methods. EDA methods such as descriptive statistics and histograms are at the core of data science, and they have been widely used for the analysis, visualization and interpretation of data [26]. For a more robust analysis of the SDN performance metrics, RSA is conducted by fitting a linear regression model to understand and statistically account for the pair-

wise interactions between the jitter (J_t), response time (R_t) and throughput (T_p) metrics of the emulated SDN when it is operating without attack ("Normal" state) and when it is subjected to HTTP DDoS flooding attacking ("HTTP" state), TCP DDoS flooding attack ("TCP" state) and UDP DDoS flooding attacking ("UDP" state). All experiments analyzed and discussed in this section have been carried out on a workstation with Intel 6-core i7-8700 3.20 GHz CPU and 32.0 GB RAM, except where stated otherwise. The elapsed times reported are elapsed real times from a wall clock.

5.1. Descriptive Statistics of the SDN Parameters

The description of the dataset generated according to the experiments carried out in Section 3 are further detailed in [15,16]. However, the descriptive statistics for T_p, R_t and J_t over the entire dataset for the emulated scenarios are reported again in Tables 1–4 to make this work self-contained.

Similar to the inferences in [16], from Tables 1–4, it can be deduced that T_p, R_t and J_t all vary according to the SDN scenarios indicating their susceptibility to the SDN scenarios. The goal is to visualize and analyze the distributions, ascertain the sensitivities and understand the level of interactions of the SDN performance metrics for the respective SDN scenarios as carried out in the following subsections.

5.2. Distributions of the SDN Parameters

To visualize the distributions of the SDN parameters under consideration for the SDN scenarios described in Section 3, histograms with distribution fits based on probability density function (PDF) are used. The visualizations are shown in Figures 4–6. The PDFs for the normal distributions in Figures 4–6 with mean (μ), standard deviation (σ), and variance (σ^2) are derived as follows:

$$f(X, \mu, \sigma) = \frac{1}{\mu\sqrt{2\pi}} exp\left[-\frac{(X-\mu)^2}{2\mu^2} \right] \tag{2}$$

From Figure 4, the following inferences can be made for T_p: (1) T_p is affected or altered when the SDN is subjected to a DDoS flooding attack. (2) The distribution of T_p did not change or vary noticeably when the SDN is subjected to UDP flooding attack. (3) The distribution of T_p changed or varied noticeably when the SDN is subjected to TCP flooding and HTTP flooding attacks with most of the metrics distributed around 0, as opposed to the range of around 95 to around 96, when the SDN is operating normally or subjected to UDP flooding attack. From a practical viewpoint, in Figure 4, T_p can be said to be more susceptible to TCP and HTTP flooding attacks due to the way in which TCP and HTTP flooding attacks work—bombardment of the target server with multiple connection requests to consume the server's network resources and inundation of the target server with multiple browser-based internet requests that will eventually cause denial-of-service to additional legitimate requests, respectively [27,28]. In a sense, T_p is typically affected in these scenarios. However, a scenario in which the targeted server utilizes resources to check and then responds to each received UDP packet, including spoofed UDP packets, as in the case of the UDP flooding attack, may not necessarily affect T_p.

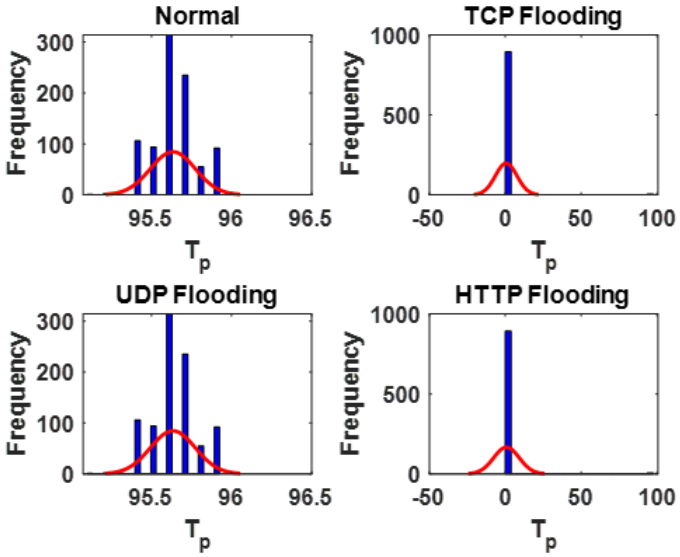

Figure 4. The distributions of T_p metrics for various SDN scenarios.

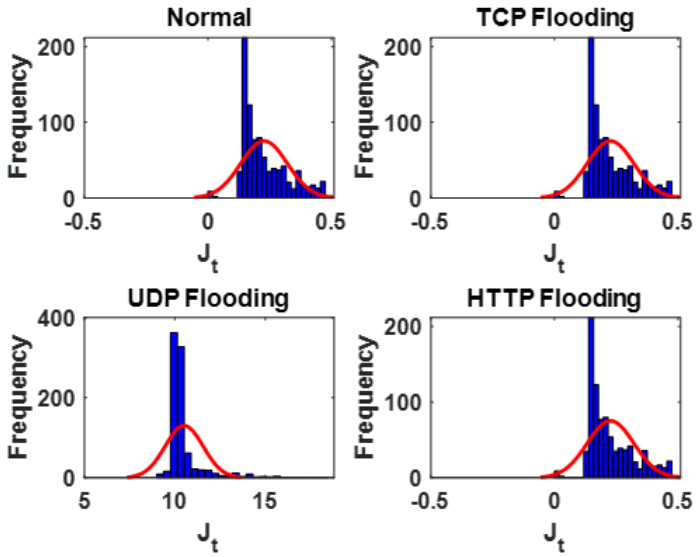

Figure 5. The distributions of J_t metrics for various SDN scenarios.

From Figure 5, the following inferences can be made for J_t: (1) J_t is affected or altered when the SDN is subjected to a DDoS flooding attack. (2) The distribution of J_t did not change or vary noticeably when the SDN is subjected to TCP flooding and HTTP flooding attacks. (3) The distribution of J_t changed or varied noticeably when the SDN is subjected to UDP flooding attack with most of the metrics distributed over the range of around to around 15, as opposed to the range of around 0 to around 0.5, when the SDN is operating normally or subjected to TCP flooding attack or subjected to HTTP flooding attack. Jitter is all about timing and the sequence of the arriving packets. If packets arrive in bursts interspersed with gaps, or if they arrive out of sequence, then jitter values will be high. From a practical viewpoint, in Figure 5, J_t can be said to be more susceptible to UDP DDoS flooding attack due to the way in which UDP DDoS flooding attacks work—a handshake is not required, and the targeted server is flooded with UDP traffic without first getting the server's permission to initiate communication [29]. In a sense, T_p and R_t are typically not affected in this scenario. However, the impact of the UDP DDoS flooding attack can be more severe when running voice over IP (VoIP) applications [30].

From Figure 6, the following inferences can be made for R_t: (1) R_t is affected or altered when the SDN is subjected to a DDoS flooding attack. (2) The distribution of R_t changed or

varied noticeably when the SDN is subjected to TCP flooding, UDP and HTTP flooding attacks. (3) Most of the metrics are distributed over a range of around 0 to around 500 for TCP flooding attack, a range of around 10 to around 50 for UDP flooding attack, and a range of around 0 to around 200 for HTTP flooding attack, in sharp contrast to a range of around 0 to around 0.5, when the SDN is operating normally. From a practical viewpoint, in Figure 6, R_t can be said to be susceptible to all the investigated DDoS flooding attacks (TCP, UDP and HTTP) due to the nature of these attacks (already discussed above) [27–29]. As a result, R_t is a critical network monitoring metric, and it can be drastically affected by DDoS flooding attacks, especially in applications that require waiting for an acknowledgement before sending any more packets. In such situations, the unified communication systems of SDNs are often hampered.

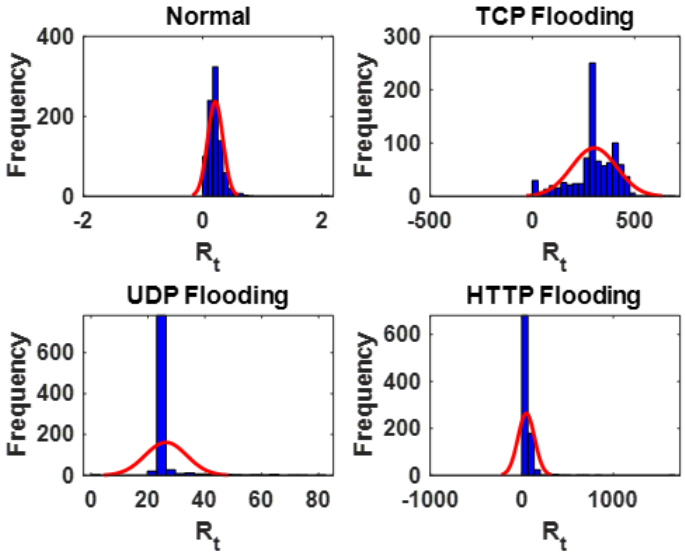

Figure 6. The distributions of R_t metrics for various SDN scenarios.

5.3. Pairwise Covariances and Correlations of the SDN Parameters

To have the measures that indicate the extent to which the SDN performance metrics (i.e., J_t, R_t and T_p) change in tandem (alongside each other), their pairwise covariances are derived to have the covariance matrices reported in Tables 5–7. A measure of covariance between any two observations (X_i and Y_i) in the dataset of the SDN performance metrics can be described mathematically as follows:

$$Cov(X_i, Y_j) = \frac{\sum_{i=1}^{n}(X_i - \mu_X)(Y_i - \mu_Y)}{n-1} \tag{3}$$

where n is the total number of observations in the dataset, i is the ith observation in the dataset, μ_X is the mean of all observations for X and μ_Y is the mean of all observations for Y such that:

$$Cov(X, X) = Var(X) = \sigma_X^2 \tag{4}$$

$$Cov(Y, Y) = Var(Y) = \sigma_Y^2 \tag{5}$$

where σ_X^2 and σ_Y^2 are the variances of all observations for X and Y, respectively.

Table 5. Covariance matrix for T_p when the SDN is operating normally and subjected to popular variants of DDoS flooding attacks.

SDN States or Scenarios	Normal	TCP	UDP	HTTP
Normal	0.0197	0.0036	0.0197	−0.0035
TCP	0.0036	50.4086	0.0036	50.3465
UDP	0.0197	0.0036	0.0197	−0.0035
HTTP	−0.0035	50.3465	−0.0035	70.4837

From Table 5, the following inferences can be drawn for T_p: (1) The covariances between the T_p metrics for the paired SDN scenarios are mostly close to or approaching zero. This suggests that the paired values of the T_p metrics may vary independently of each other for these paired SDN scenarios, except for when the SDN is subjected to TCP flooding attack and HTTP flooding attack. This corroborates the analysis of the distributions of T_p discussed in Section 5.2. (2) The large positive covariance between the T_p metrics for when the SDN is subjected to TCP flooding attack and HTTP flooding attack indicates that, when the T_p metric resulting from subjecting the SDN to TCP flooding attack is above its mean, the T_p metric from subjecting the SDN to HTTP flooding attack will probably also be above its mean, and vice versa. In other words, the paired values of the T_p metric for both scenarios tend to increase together.

Table 6. Covariance matrix for J_t when the SDN is operating normally and subjected to popular variants of DDoS flooding attacks.

SDN States or Scenarios	Normal	TCP	UDP	HTTP
Normal	0.0089	0.0089	−0.0022	0.0089
TCP	0.0089	0.0089	−0.0022	0.0089
UDP	−0.0022	−0.0022	1.1017	−0.0022
HTTP	0.0089	0.0089	−0.0022	0.0089

From Table 6, the following inferences can be drawn for J_t: (1) The covariances between the J_t metrics for the SDN scenarios are mostly close to or approaching zero. This suggests that the paired values for J_t metrics may vary independently of each other for the paired SDN scenarios. This corroborates the analysis of the distributions of J_t discussed in Section 5.2. (2) The negative covariances between the J_t metrics when the SDN is subjected to UDP flooding attack and when it is operating normally or subjected to TCP flooding attack or subjected to HTTP flooding attack indicate that, when the J_t metric is above its mean in any one of these paired SDN scenarios, it is probably below its mean in any one of the other SDN scenarios in the same pair. In other words, an inverse relationship exists between the J_t metrics for these paired SDN scenarios. Note that, since the negative covariances are close to or approaching zero, the first inference could be a better generalized inferential assessment.

Table 7. Covariance matrix for R_t when the SDN is operating normally and subjected to popular variants of DDoS flooding attacks.

SDN States or Scenarios	Normal	TCP	UDP	HTTP
Normal	0.0165	−2.2539	0.1729	0.0314
TCP	−2.2539	1.2246×10^4	-3.2816×10^2	-1.3091×10^3
UDP	0.1729	-3.2816×10^2	5.3648×10^1	6.4838×10^1
HTTP	0.0314	-1.3091×10^3	6.4838×10^1	8.2701×10^3

From Table 7, the following inferences can be drawn for R_t: (1) The covariances between the J_t metrics for the paired SDN scenarios are mostly positive and negative. This suggests the that R_t metrics for these paired SDN scenarios tend to be either above or below

their means. (2) The approximately null covariance between the R_t metrics for when the SDN is operating normally and, when it is subjected to HTTP flooding attack, suggests that the paired values of the R_t metrics may vary independently of each other. (3) The positive covariance between the R_t metrics when the SDN is subjected to UDP flooding attack and when it is subjected to HTTP flooding or operating normally suggests that, when the R_t metric is above its mean in any one of these paired SDN scenarios, it is probably above its mean in any one of the other SDN scenarios in the same pair. In other words, the paired values of the J_t metrics increase together for these paired SDN scenarios. (4) The negative covariance between the R_t metrics when the SDN is subjected to TCP flooding attack and when it is operating normally or subjected to UDP flooding attack or subjected to HTTP flooding attack indicates that, when the J_t metric is above its mean in any one of these paired SDN scenarios, it is probably below its mean in any one of the other SDN scenarios in the same pair. In other words, an inverse relationship exists between the R_t metrics for these paired SDN scenarios.

Covariances tend to be difficult to interpret, so measures of correlations are often required for more robust analysis [26]. For example, covariances close to or approaching null for X_i and Y_i indicates that X_i and Y_i vary independently of each other. This can be observed in Tables 5–7 for T_p, J_t and R_t, respectively. In a technical sense, independence implies correlation, but the converse or reverse is not necessarily true. To have a scaled form of the covariance measures reported in Tables 5–7, pairwise correlation measures that represent how strongly the SDN performance metrics are related to each other are derived to have the correlation matrix reported in Tables 8–10. A measure of correlation between any two sets of observations (X and Y) in the dataset of the SDN performance metrics can be described mathematically as follows:

$$r_{X,Y} = \frac{Cov(X,Y)}{\mu_X \mu_Y} \tag{6}$$

Table 8. Correlation matrix for T_p when the SDN is operating normally and subjected to popular variants of DDoS flooding attacks.

SDN States or Scenarios	Normal	TCP	UDP	HTTP
Normal	1.0000	0.0036	1.0000	−0.0030
TCP	0.0036	1.0000	0.0036	0.84460
UDP	1.0000	0.0036	1.0000	−0.0030
HTTP	−0.0030	0.8446	−0.0030	1.0000

From Table 8, the following inferences can be drawn for T_p: (1) The correlations between the T_p metrics are mostly positive and approaching null, indicating that some linear positive relationships exist between the T_p metrics for the SDN scenarios, but they are not necessarily strong. (2) A perfect positive linear correlation exists between the T_p metrics for when the SDN is operating normally and when it is subjected to UDP flooding attack. This corroborates the analysis of the distributions of T_p metrics discussed in Section 5.2. (3) A strong positive linear correlation exists between the T_p metrics for when the SDN is subjected to HTTP flooding attack and when it is subjected to UDP flooding attack. This also corroborates the analysis of the distributions of T_p metrics discussed in Section 5.2. (4) Negative linear correlations exist between the T_p metrics for when the SDN is operating normally and when it is subjected to HTTP flooding, and for when the SDN is subjected to UDP flooding attack and HTTP flooding attack.

Table 9. Correlation matrix for J_t when the SDN is operating normally and subjected to popular variants of DDoS flooding attacks.

SDN States or Scenarios	Normal	TCP	UDP	HTTP
Normal	1.0000	1.0000	−0.0227	1.0000
TCP	1.0000	1.0000	−0.0227	1.0000
UDP	−0.0227	−0.0227	1.0000	−0.0227
HTTP	1.0000	1.0000	−0.0227	1.0000

From Table 9, the following inferences can be drawn for J_t: (1) The correlations between the J_t metrics are mostly positive and unity, indicating that perfect linear positive relationships exist between the J_t metrics for the SDN scenarios. (2) Some of the correlations between the J_t metrics are negative and approaching null, indicating that some negative linear positive relationships exist between the J_t metrics for the SDN scenarios, but they are not necessarily strong. (3) Perfect positive linear correlations exist between the J_t metrics for when the SDN is operating normally and when it is subjected to TCP flooding attack or HTTP flooding attack, and for when the SDN is subjected to TCP flooding attack and HTTP flooding attack. This corroborates the analysis of the J_t metrics carried out earlier. (4) Negative linear correlations exist between the J_t metrics for when the SDN is operating normally and when it is subjected to UDP flooding attack, and for when the SDN is subjected to UDP flooding attack and TCP flooding attack or HTTP flooding attack. This also corroborates the analysis of distributions of the J_t metrics discussed in Section 5.2.

Table 10. Correlation matrix for R_t when the SDN is operating normally and subjected to popular variants of DDoS flooding attacks.

SDN States or Scenarios	Normal	TCP	UDP	HTTP
Normal	1.0000	−0.1584	0.1836	0.0027
TCP	−0.1584	1.0000	−0.4049	−0.1301
UDP	0.1839	−0.4049	1.0000	0.0973
HTTP	0.0027	−0.1301	0.0973	1.0000

From Table 10, the following inferences can be drawn for R_t: (1) Some of the correlations between the R_t metrics are positive and approaching null, indicating that some linear positive relationships exist between the R_t metrics for the SDN scenarios, but they are not necessarily strong. (2) Some of correlations between the J_t metrics are negative, indicating that some negative linear positive relationships exist between the R_t metrics for the SDN scenarios. (3) Significant positive linear correlations exist between the R_t metrics for when the SDN is operating normally and when it is subjected to UDP flooding attack. This corroborates the analysis of the distributions of J_t metrics discussed in Section 5.2. (4) Significant negative linear correlations exist between the R_t metrics for when the SDN is operating normally and when it is subjected to TCP flooding attack, and for when the SDN is subjected to TCP flooding attack and when it is subjected to UDP flooding attack or HTTP flooding attack. This also corroborates the analysis of the distributions of J_t metrics discussed in Section 5.2.

The single run of the EDA implementation costs about 3.2 s in total on the workstation mentioned above. Thus, it can be said, in practice, a suitable augmentation of the EDA process that features as a complementary add-on or toolbox to existing SDN monitoring and evaluation software will likely offer a promising data analytic framework for real-time trend analysis, pattern recognition and overall monitoring and evaluation of SDN traffic and performance metrics to an order of less than 5 s on conventional workstations, for every 15-min samples of SDN data collected. This time window is sufficient and reasonably short, and it will ultimately reduce the current average time required to respond to potential attacks on real-world SDNs because inferences can be made in a shorter time.

5.4. Regression-Based Sensitivity Analysis of the SDN Parameters

Regression analysis is a popular and widely used data-driven methodology [31]. It is a form of supervised learning, and it is typically implemented as linear regression analysis in the conventional analysis of data such as SDN data [32]. Primarily, it allows for a methodical and mathematical way of sorting the impact of the independent variable(s) on the associated dependent variable in each given dataset. In doing so, the significance of the independent variable(s) and the interactions between them relative to the dependent variable can be understood as discussed in the following subsections for the SDN data in this work.

5.4.1. Feature Scaling and Feature Engineering

Feature scaling and feature engineering are very common data pre-processing procedures in the implementation of machine learning [33,34] when supervised learning techniques such as classification and/or regression are employed. Feature scaling mainly involves the normalization of the range of the features or independent variables in the dataset to ensure that each feature or independent variable contributes relative similar numerical weights (approximately proportionally) to the targets [34]. Feature engineering on the other hand involves the transformation of raw datasets into datasets that are characterized with informative features having high predictive power [33]. Depending on the nature of the raw datasets being analyzed and the intended machine learning process, feature scaling and feature engineering can be implemented in a number of ways [33,34]. In this work, feature scaling and feature engineering have been implemented according to Equations (7) and (8) [15], respectively:

$$z_i = \frac{(X_i - \mu_X)}{\sigma_X} \tag{7}$$

where z_i is the z-score of the ith observation in a given data sample, X, having a mean and standard deviation of μ_X and σ_X, respectively:

$$AURV = (T_p^{o_i,z} - (J_t^{o_i,z} + R_t^{o_i,z})) \times w \tag{8}$$

where $w \in (0,1)$ is a random weight that is uniformly distributed, and it penalizes $AURV$ based on the operating condition that the ith SDN event or scenario is linked to. The values of w are deduced as follows:

$$w = \begin{cases} 0 < w < 0.5; \text{ If SDN scenario is 'Normal'.} \\ 0.5 < w < 1; \text{Otherwise.} \end{cases} \tag{9}$$

Equation (7) ensures that T_p, J_t and R_t metrics all have a zero-mean (when subtracting μ_{T_p}, μ_{J_t}, and μ_{R_t}, respectively) and unit-variance. It also ensures that the gradient descent of the linear regression implementation moves smoothly towards the minima and that the gradient descent steps are updated at a similar rate for the T_p, J_t and R_t metrics prior to the linear regression implementation (see Section 5.4.2). This is because machine learning techniques such as linear regression that use gradient descent for the minimization of the loss function are highly sensitive to the range of the input features as mathematically described in the next subsection.

A multivariate visualization of the standardized metrics for T_p, J_t, and R_t and the resulting values of $AURV$ across all the SDN scenarios are reported using a parallel coordinates plot shown in Figure 7. From Figure 7, it can be inferred that the relative numeric weights of standardized T_p, J_t and R_t metrics are proportionate in their contributions to the values of $AURV$. Note that the derivation in Equation (8) allows for the creation of an artificial boundary between the normal SDN scenario (normal operating state of the SDN) and other SDN scenarios (TCP, UDP and HTTP SDN scenarios—states in which the SDN is subjected to TCP flooding, UDP flooding and HTTP flooding DDoS attacks, respectively)

as verified in [15] and revealed again in Figure 8 to make this work self-contained; hence, it is adopted in this work. However, in contrast to the work carried out in [15], where *AURV* is used as a feature or independent variable to address a classification problem; it is introduced and used as a mathematical cost function to reflect scenario-specific targets for all states of the emulated SDN in terms of the performance metrics (i.e., T_p, J_t and R_t) in this study. Note that, for the parametric study of SDNs via supervised learning, the derivation of mathematical cost functions to synthetically generate the targets is not unconventional, and it has recently been typified in [16].

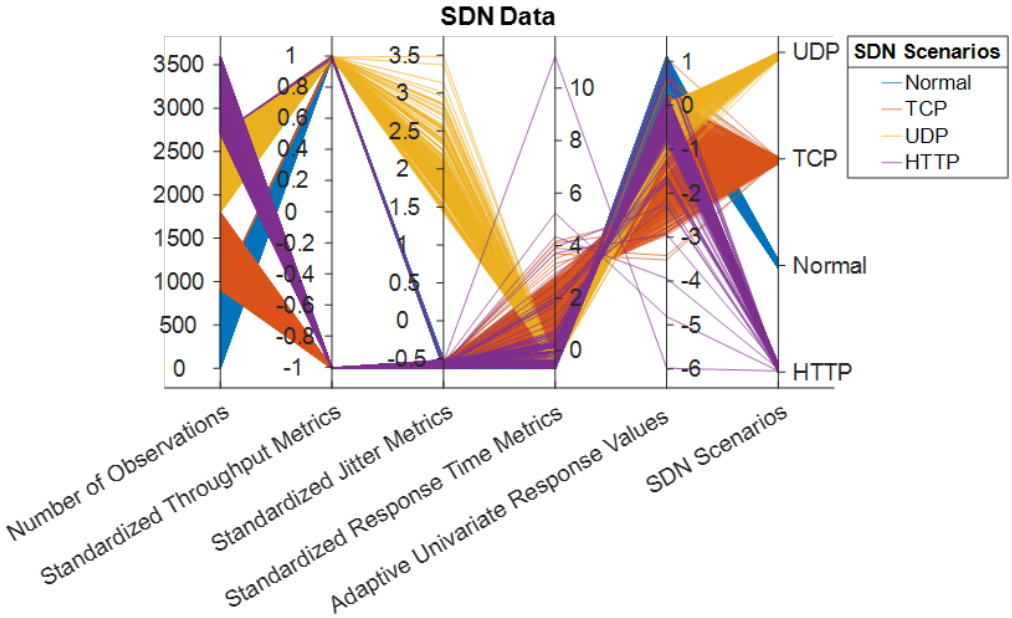

Figure 7. Parallel coordinates of the zero scores of T_p, J_t and R_t metrics for all the SDN scenarios.

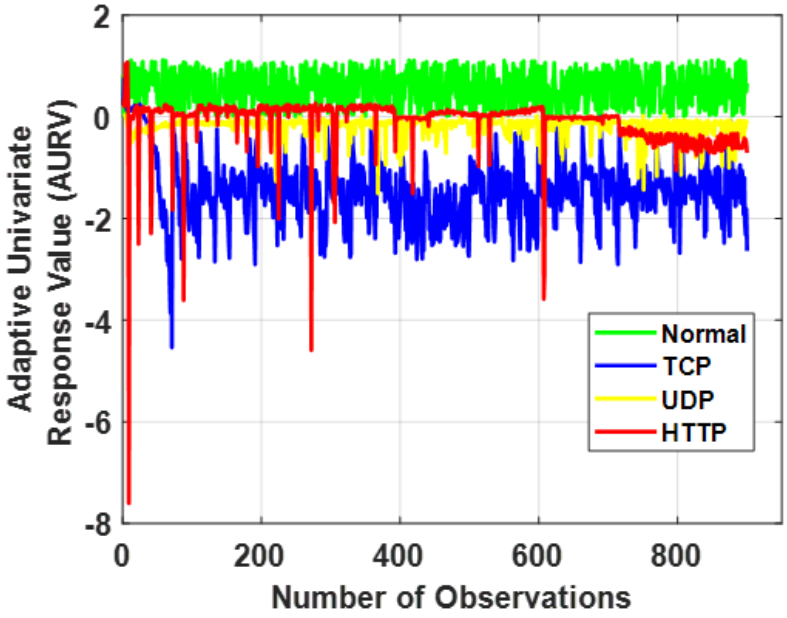

Figure 8. Trend of the AURVs for T_p, J_t and R_t metrics for all the SDN scenarios.

5.4.2. Regression Analysis

For the sensitivity analysis of the SDN dataset, supervised learning (linear regression) is carried out to ascertain the interactions between the performance metrics (i.e., T_p, J_t and R_t) as the SDN is being subjected to DDoS flooding attacks (i.e., as *AURV* changes

according to changes in the ordered triads or combinations of T_p, J_t and R_t for instances on the SDN). Note that, with this approach, the standardized regression coefficients are employed directly to evaluate sensitivity [35]. The linear regression implementation can be mathematically described as follows:

$$AURV_i = f(T_p^i, J_t^i, R_t^i, \beta) + e_i \tag{10}$$

where $AURV_i$, T_p^i, J_t^i, R_t^i, β and e_i are the $AURV$, T_p, T_t and R_t, unknown parameters (typically, scalar coefficients) and error terms (typically, scalar), respectively, for the ith observation in the SDN data set. Equation (10) is then used to predict $AURV$ for new or arbitrary values of T_p, J_t and R_t.

In terms of computational complexity, linear regression implementation is generally aimed at solving the linear or matrix algebra problem described as follows:

$$(AA')^{-1} \times A'B \tag{11}$$

where A is the explanatory or predictor or independent variable, and it is a (3600×3) matrix as discussed in Section 3, A' is the transpose of A, and B is the response or target or independent variable, and it is a (3600×1) matrix. As a result, A holds all the T_p, T_t and R_t metrics for all the SDN scenarios investigated, and B holds the corresponding $AURV$ values for the metrics.

Considering Equation (11) and the dimensions of A and B, respectively, the matrix product $A \times A'$ will have a complexity of $\mathcal{O}(3^2 \times 3600)$, the matrix–vector product $A' \times B$ will have a complexity of $\mathcal{O}(3 \times 3600)$, and the inverse operation $A \times A'^{-1}$ will have a complexity of $\mathcal{O}(3^3)$. Hence, the overall complexity of the linear regression implementation becomes $\mathcal{O}(3600 \times 3^2 + 3^3)$, such that, for any SDN_n total number observations on the SDN having any SDN_p metrics, the complexity may be generalized and estimated as follows:

$$\mathcal{O}(SDN_n \times SDN_p^2 + SDN_p^3) \tag{12}$$

where SDN_n and SDN_p assume their values according to the given linear regression problem.

To build the linear regression model, the built-in function "fitlm" in MATLAB's statistics and machine learning toolbox is used [36], and it costs about 3.3 s on the workstation described above. The algorithmic framework of "fitlm" premises on QR decomposition or factorization and the use of M-estimation for robustness [37]. M-estimation formulates estimating equations and solves them using the iteratively re-weighted least squares (IRLS) method [38]. The linear regression model built as a result can be mathematically described as follows:

$$AURV \approx 1 + T_p \times J_t + T_p \times R_t + J_t \times R_t \tag{13}$$

Equation (13) corresponds to:

$$AURV = \beta_0 + \beta_1 \times T_p + \beta_2 \times J_t + \beta_3 \times R_t + \epsilon \tag{14}$$

where β_0, β_1, β_2 and β_3 are the coefficients and ϵ is the error term. The coefficient of determination (R-squared value) of the regression model is then deduced as follows:

$$1 - \frac{SSE}{SST} = 0.926 \tag{15}$$

where SSE is the sum of squared estimate of errors (i.e., sum of the predicted deviations from the actual empirical values of the data, also known as the sum of the squares of residuals), and SST is the sum of squares total or total sum of squares (the sum over all squared differences between the observations and their overall mean). An R-squared value (R^2) of 0.926 for the regression model also indicates that 92.6% of response variable variation is explained by the regression model. This relatively large value of R^2 indicates

that the model's response is in fact linear and the approach of using the standardized regression coefficients as direct measures of sensitivity is suitable [39,40].

To further analyse the linear regression model, its root mean square of error (RMSE) is deduced as follows:

$$\sqrt{\frac{\sum_{i=1}^{n}\left(AURV_i - AURV_i^{'}\right)^2}{n}} = 0.235 \quad (16)$$

where $AURV_i$ and $AURV_i^{'}$ are the actual and predicted $AURV$ for the ith SDN observation among all the SDN observations (i.e., $n = 3600$) visualized in Figure 8. An RMSE value of 0.235 (low and tending towards to null) indicates that the linear regression model is of a high quality [41].

Since the regression analysis involves multiple predictors or explanatory variables (i.e., T_p, T_t, and R_t), an added variable plot shown in Figure 9 is used to visualize the transformations of $AURV$ that nets out the influence of all transformations of T_p, J_t and R_t (i.e., the whole model) as recommended in [42]. The chosen level of statistical confidence is 95%-based such that, if the confidence interval excludes a null slope, the model is likely to be statistically significant [42]. Figure 9 shows that the linear regression model is significant because a horizontal line does not fit between the confidence bounds, as revealed in the zoomed in section of Figure 9.

The summary of the estimated coefficients and statistics for the linear regression model are detailed in Table 11. According to the p-Values (all $\lll 0.05$) for the t-statistic of the hypothesis test that the corresponding coefficient is equal to zero or not in Table 11, $AURV$ (the response or target variable) differs significantly according to T_p, J_t and R_t metrics (the explanatory variables) and the pairwise interactions between them at the 5% significance level or 95% confidence level. Thus, all the SDN metrics and the pairwise interactions between them are significant.

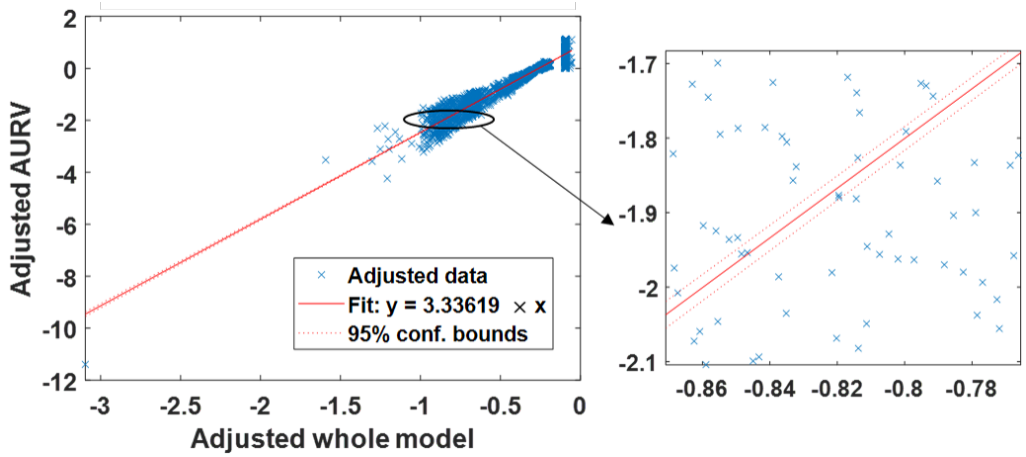

Figure 9. Added-variable plot for the linear regression model.

The main and interaction effects of the linear regression model in Figure 9 and described in Table 11 are explored to characterize the sensitivity of T_p, T_t and R_t relative to the changing SDN scenarios numerically factored into $AURV$. The plots of the main effect, interaction effect between T_p and J_t, interaction effect between T_p and R_t and interaction effect between R_t and J_t are shown in Figures 10–13, respectively.

Table 11. Summary of estimated coefficients and statistics for the regression analysis.

Variables or Parameters	Estimate	Standard Error	t-Statistic	*p*-Value
Intercept	0.8684	0.1298	6.6904	2.5721×10^{-11}
T_p	1.2461	0.0988	12.6090	1.0575×10^{-35}
J_t	−0.81759	0.1304	−6.2695	4.0527×10^{-10}
R_t	0.3714	0.0936	3.9665	7.4360×10^{-5}
$T_p : J_t$	−0.7082	0.1604	−4.4140	1.0448×10^{-5}
$T_p : R_t$	2.1865	0.1657	13.1920	7.6537×10^{-39}
$J_t : R_t$	−1.8678	0.1555	−12.0090	1.3344×10^{-32}

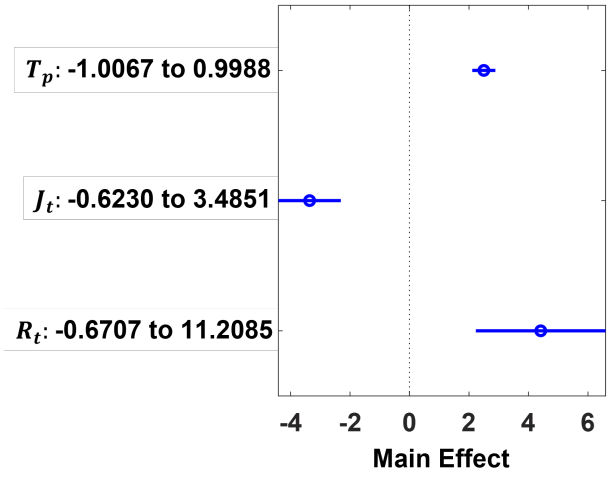

Figure 10. Main effect: T_p, J_t, and R_t metrics for all the SDN scenarios.

From Figure 10, the following inferences can be made: (1) An increase in T_p from −1.0067 to 0.9988 causes an expected 3-unit increase in $AURV$, given all else held constant. (2) An increase in J_t from −0.6230 to 3.4851 causes an expected 3-unit decrease in $AURV$, given all else held constants. (3) An increase in R_t from −0.6707 to 11.2085 causes an expected increase of over four units in $AURV$, given all else held constants. (4) T_p, J_t and R_t metrics are all sensitive to changes in the SDN scenarios as inferred by the main effects on $AURV$. From a practical viewpoint, it can be said that, as the SDN transitions from its normal operating state to other states because of DDoS flooding attacks, its key performance metrics (i.e., T_p, J_t, and R_t) fluctuate with some level of interactions between them for every change that they undergo. This can be expected due to the intrinsic interdependence between the performance metrics. For example, R_t is related to the latency of the SDN, and J_t is a factor of the change in latency of the SDN.

From Figures 11–13, the following inference can be made: (1) For each of the explanatory variables or predictors (i.e., T_p and J_t) in Figure 11, the main effect point (blue circle as in Figure 10) and the conditional effect points (red circles showing the impact of varying T_p and J_t) are not all exactly vertically aligned. This indicates the existence of interaction effects on the response variable (i.e., $AURV$). This corroborates the p-value of T_p:J_t in Table 11, 1.0448×10^{-5}, which is much lower than 0.05. (2) For each of the explanatory variables or predictors (i.e., T_p and R_t) in Figure 12, the main effect point (blue circle as in Figure 10) and the conditional effect points (red circles showing the impact of varying T_p and R_t) are not all exactly vertically aligned. This indicates the existence of interaction effects on the response variable (i.e., $AURV$). This corroborates the p-value of T_p:R_t in Table 11, 7.6537×10^{-39}, which is much lower than 0.05. (3) For each of the explanatory variables or predictors (i.e., J_t and R_t) in Figure 13, the main effect point (blue circle as in Figure 10) and the conditional effect points (red circles showing the impact of varying T_p and R_t) are not all exactly vertically aligned. This indicates the existence of interaction

effects on the response variable (i.e., *AURV*). This corroborates the p-value of T_p:R_t in Table 11, 1.3344×10^{-32}, which is much lower than 0.05.

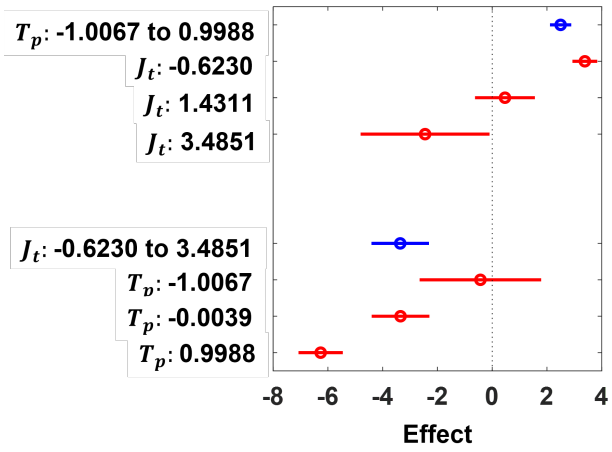

Figure 11. Interaction effect: T_p and J_t metrics for all the SDN scenarios.

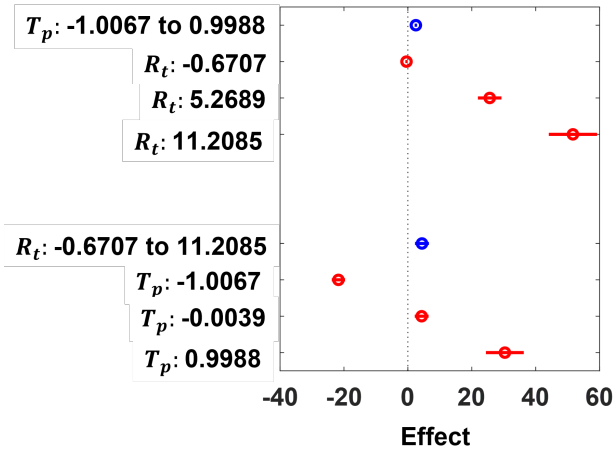

Figure 12. Interaction effect: T_p and R_t metrics for all the SDN scenarios.

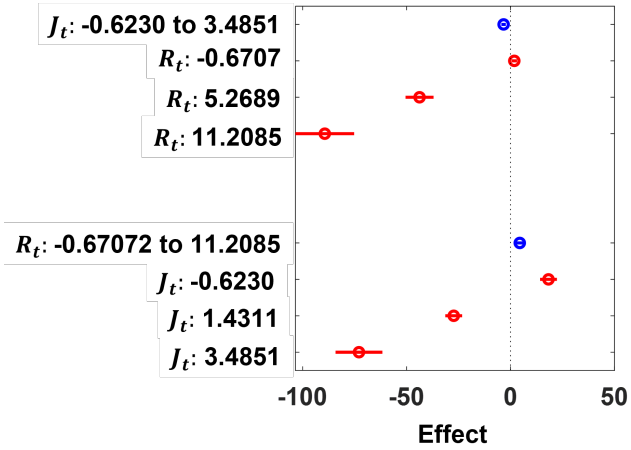

Figure 13. Interaction effect: J_t and R_t metrics for all the SDN scenarios.

6. Guidelines for Inference-Based Assessment of SDN

In this section, guidelines are provided for the inference-based assessment of SDNs using T_p, J_t and R_t metrics. Specifically, in Table 12, a summarized qualitative assessment of SDN based on the EDA and regression analysis carried out in Section 5.4.2 of this article is outlined. These guidelines will prove useful for network operators and administrators in characterizing the distribution of the considered flooding attacks—thus shortening the response time to handle these attacks in typical SDN environments.

Table 12. Summarized EDA-based qualitative assessment of SDN metrics.

Metric	TCP Flooding	HTTP Flooding	UDP Flooding
T_p	Significant mean shift and deviation from the "normal" scenario (mean zeroing), and significant expansion of distribution spread.	Insignificant deviation from the normal scenario	Significant mean shift and deviation from the "normal" scenario (mean zeroing), and significant expansion of distribution spread.
J_t	Insignificant deviation from the "normal" scenario.	Significant (positive) shifting of the mean of the distribution.	Insignificant deviation from the "normal" scenario.
R_t	Significant (positive) expansion of the spread of the distribution, and significant (positive) mean shift.	Slight (positive) mean shift and expansion of the distribution spread.	Significant expansion of the spread and mean zeroing.

From Table 12 above, it can be inferred that the DDoS flooding attack will likely be a TCP flooding attack when the distributions of T_p and R_t experience a significant (positive) shift from the "normal" state and a significant expansion of the distribution spread with the distribution of J_t remaining unaffected. Similarly, a UDP DDoS flooding attack can be inferred when the distribution of T_p experiences insignificant deviation from the "normal" state, whereas both J_t and R_t experience mean shifts and distribution spread expansion. However, the mean shift in the distribution of R_t is slight and that of J_t is significant. From the RSA, T_p, J_t and R_t metrics of the SDN are all sensitive to the respective DDoS flooding attacks. Tables 13 and 14 provide a summarized RSA-based qualitative assessment of the sensitivities and pairwise interactions of the studied network parameters (i.e., T_p, J_t, and R_t metrics), respectively.

Table 13. Summarized RSA-based qualitative assessment of SDN metrics (sensitivities).

Metric	Comment
T_p	Sensitive.
J_t	Sensitive.
R_t	Very sensitive.

Table 14. Summarized RSA-based qualitative assessment of SDN metrics (pairwise interactions).

Pair	Comment
T_p:J_t	Significant interaction.
T_p:R_t	Most significant interaction.
J_t:R_t	Very significant interaction.

7. Conclusions

EDA and RSA have been employed to undertake a behavioral study of SDN parameters (T_p, J_t and R_t) given emulated SDN scenarios that are representative of popular real-world SDN events or states (normal operating condition without any incidents on the SDN, and hypertext transfer protocol (HTTP) flooding, transmission control protocol (TCP) flooding and user datagram protocol (UDP) flooding, when the SDN is subjected to a distributed denial-of-service (DDoS) attack). As a proposed methodology in this paper, the behavioral study reveals that the trends or distributions of T_p, J_t and R_t performance metrics on the SDN vary according to given SDN events or states via EDA, and the succeeding RSA ascertains the existence and level of pairwise interactions between the SDN performance metrics (T_p, J_t and R_t) used to evaluate the emulated SDN scenarios, both validating the proposed methodology. The findings of the EDA and RSA carried out are summarized to provide SDN administrators and operators with inference-based guidelines for the appraisal of SDNs. Even though these guidelines are not exhaustive, they are expected to be sufficient in informing SDN administrators and operators about the likelihood of an attack on the SDN based on analysis and visualization of the SDN performance metrics. In the future, real-world on-the-fly SDN data will be used to corroborate the investigations and findings in this work as a backbone for the development of full-fledged guidelines for SDN administrators and operators.

Author Contributions: Conceptualization, M.O.A.; Data curation, M.O.A. and A.O.S.; Formal analysis, M.O.A., A.O.S. and U.E.U.; Investigation, M.O.A. and A.O.S.; Methodology, M.O.A.; Software, M.O.A.; Validation, M.O.A., A.O.S., and U.E.U.; Writing—original draft, M.O.A., A.O.S. and U.E.U.; Writing—review & editing, M.O.A., A.O.S. and U.E.U. All authors have read and agreed to the published version of the manuscript.

Funding: The APC for this work was supported in part by the Faculty of Arts, Science and Technology, Wrexham Glyndŵr University, UK.

Data Availability Statement: SDN Dataset for DDoS Flooding Attack Detection. Available at: Kaggle (https://doi.org/10.34740/KAGGLE/DSV/3965784, accessed on 30 May 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kumar, R.; Swarnkar, M.; Singal, G.; Kumar, N. IoT Network Traffic Classification Using Machine Learning Algorithms: An Experimental Analysis. *IEEE Internet Things J.* **2022**, *9*, 989–1008. [CrossRef]
2. Ghosh, S.; Dagiuklas, T.; Iqbal, M.; Wang, X. A Cognitive Routing Framework for Reliable Communication in IoT for Industry 5.0. *IEEE Trans. Ind. Inform.* **2022**, *18*, 5446–5457. [CrossRef]
3. Zong, L.; Memon, F.H.; Li, X.; Wang, H.; Dev, K. End-to-End Transmission Control for Cross-Regional Industrial Internet of Things in Industry 5.0. *IEEE Trans. Ind. Inform.* **2022**, *18*, 4215–4223. [CrossRef]
4. Aebersold, S.A.; Akinsolu, M.O.; Monir, S.; Jones, M.L. Ubiquitous Control of a CNC Machine: Proof of Concept for Industrial IoT Applications. *Information* **2021**, *12*, 529. [CrossRef]
5. Skripcak, T.; Tanuska, P.; Konrad, U.; Schmeisser, N. Toward Nonconventional Human–Machine Interfaces for Supervisory Plant Process Monitoring. *IEEE Trans. Hum.-Mach. Syst.* **2013**, *43*, 437–450. [CrossRef]
6. Schulz, P.; Matthe, M.; Klessig, H.; Simsek, M.; Fettweis, G.; Ansari, J.; Ashraf, S.A.; Almeroth, B.; Voigt, J.; Riedel, I.; et al. Latency Critical IoT Applications in 5G: Perspective on the Design of Radio Interface and Network Architecture. *IEEE Commun. Mag.* **2017**, *55*, 70–78. [CrossRef]
7. Ferrari, P.; Flammini, A.; Sisinni, E.; Rinaldi, S.; Brandão, D.; Rocha, M.S. Delay Estimation of Industrial IoT Applications Based on Messaging Protocols. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 2188–2199. [CrossRef]
8. Nisar, K.; Jimson, E.R.; Hijazi, M.H.A.; Welch, I.; Hassan, R.; Aman, A.H.M.; Sodhro, A.H.; Pirbhulal, S.; Khan, S. A survey on the architecture, application, and security of software defined networking: Challenges and open issues. *Internet Things* **2020**, *12*, 100289. [CrossRef]
9. Du, M.; Wang, K. An SDN-Enabled Pseudo-Honeypot Strategy for Distributed Denial of Service Attacks in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2020**, *16*, 648–657. [CrossRef]
10. The Shocking DDoS Attack Statistics That Prove You Need Protection. Available online: https://www.infosecurity-magazine.com/blogs/ddos-attacks-stats-protection/ (accessed on 31 January 2022).

11. Cao, Y.; Gao, Y.; Tan, R.; Han, Q.; Liu, Z. Understanding Internet DDoS Mitigation from Academic and Industrial Perspectives. *IEEE Access* **2018**, *6*, 66641–66648. [CrossRef]
12. Matta, V.; Di Mauro, M.; Longo, M. DDoS Attacks With Randomized Traffic Innovation: Botnet Identification Challenges and Strategies. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1844–1859. [CrossRef]
13. Aladaileh, M.A.; Anbar, M.; Hasbullah, I.H.; Chong, Y.W.; Sanjalawe, Y.K. Detection Techniques of Distributed Denial of Service Attacks on Software-Defined Networking Controller—A Review. *IEEE Access* **2020**, *8*, 143985–143995. [CrossRef]
14. Cirillo, M.; Mauro, M.D.; Matta, V.; Tambasco, M. Botnet Identification in DDoS Attacks With Multiple Emulation Dictionaries. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 3554–3569. [CrossRef]
15. Sangodoyin, A.O.; Akinsolu, M.O.; Pillai, P.; Grout, V. Detection and Classification of DDoS Flooding Attacks on Software-Defined Networks: A Case Study for the Application of Machine Learning. *IEEE Access* **2021**, *9*, 122495–122508. [CrossRef]
16. Sangodoyin, A.O.; Akinsolu, M.O.; Awan, I. A deductive approach for the sensitivity analysis of software defined network parameters. *Simul. Model. Pract. Theory* **2020**, *103*, 102099. [CrossRef]
17. Jhaveri, R.H.; Tan, R.; Ramani, S.V. Real-time QoS-aware Routing Scheme in SDN-based Robotic Cyber-Physical Systems. In Proceedings of the 2019 IEEE 5th International Conference on Mechatronics System and Robots (ICMSR), Singapore, 3–5 May 2019; pp. 18–23. [CrossRef]
18. Numan, P.E.; Yusof, K.M.; Marsono, M.N.B.; Yusof, S.K.S.; Fauzi, M.H.B.M.; Nathaniel, S.; Onwuka, E.N.; Baharudin, M.A.B. On the latency and jitter evaluation of software defined networks. *Bull. Electr. Eng. Inform.* **2019**, *8*, 1507–1516. [CrossRef]
19. Hossain, M.A.; Sheikh, M.N.A.; Halder, M.; Biswas, S.; Arman, M.A.I. Quality of Service in Software Defined Networking. *Glob. J. Comput. Sci. Technol.* **2018**, *18*, 20–28.
20. Goransson, P.; Black, C.; Culver, T. *Software Defined Networks: A Comprehensive Approach*; Morgan Kaufmann: Burlington, MA, USA, 2016.
21. Kreutz, D.; Ramos, F.; Verissimo, P. Towards secure and dependable software-defined networks. In Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, Hong Kong, China, 16 August 2013; ACM: New York, NY, USA, pp. 55–60.
22. Muelas, D.; Ramos, J.; de Vergara, J.E.L. Assessing the Limits of Mininet-Based Environments for Network Experimentation. *IEEE Netw.* **2018**, *32*, 168–176. [CrossRef]
23. Dos Reis Fontes, R.; Campolo, C.; Esteve Rothenberg, C.; Molinaro, A. From Theory to Experimental Evaluation: Resource Management in Software-Defined Vehicular Networks. *IEEE Access* **2017**, *5*, 3069–3076. [CrossRef]
24. Erel, M.; Teoman, E.; Özçevik, Y.; Seçinti, G.; Canberk, B. Scalability analysis and flow admission control in mininet-based SDN environment. In Proceedings of the 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), San Francisco, CA, USA, 18–21 November 2015; pp. 18–19. [CrossRef]
25. Kamaldeep.; Malik, M.; Dutta, M. Contiki-based mitigation of UDP flooding attacks in the Internet of things. In Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 5–6 May 2017; pp. 1296–1300. [CrossRef]
26. Seltman, H.J. *Experimental Design and Analysis*; Carnegie Mellon University: Pittsburgh, PA, USA, 2012.
27. Sahi, A.; Lai, D.; Li, Y.; Diykh, M. An Efficient DDoS TCP Flood Attack Detection and Prevention System in a Cloud Environment. *IEEE Access* **2017**, *5*, 6036–6048. [CrossRef]
28. Nashat, D.; Khairy, S. Detecting Http Flooding Attacks Based on Uniform Model. In Proceedings of the 2021 International Conference on Networking and Network Applications (NaNA), Lijiang, China, 29 October–1 November 2021; pp. 94–98. [CrossRef]
29. Shen, Z.Y.; Su, M.W.; Cai, Y.Z.; Tasi, M.H. Mitigating SYN Flooding and UDP Flooding in P4-based SDN. In Proceedings of the 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS), Tainan, Taiwan, 8–10 September 2021; pp. 374–377. [CrossRef]
30. Sengar, H.; Wang, H.; Wijesekera, D.; Jajodia, S. Fast Detection of Denial-of-Service Attacks on IP Telephony. In Proceedings of the 2006 14th IEEE International Workshop on Quality of Service, New Haven, CT, USA, 19–21 June 2006; pp. 199–208. [CrossRef]
31. Liu, P.; Lv, N.; Chen, K.; Tang, L.; Zhou, J. Regression Based Dynamic Elephant Flow Detection in Airborne Network. *IEEE Access* **2020**, *8*, 217123–217133. [CrossRef]
32. Shohani, R.B.; Mostafavi, S.A. Introducing a New Linear Regression Based Method for Early DDoS Attack Detection in SDN. In Proceedings of the 2020 6th International Conference on Web Research (ICWR), Tehran, Iran, 22–23 April 2020; pp. 126–132. [CrossRef]
33. Nargesian, F.; Samulowitz, H.; Khurana, U.; Khalil, E.B.; Turaga, D.S. Learning Feature Engineering for Classification. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence IJCAI, Melbourne, Australia, 19–25 August 2017; pp. 2529–2535.
34. Wan, X. Influence of feature scaling on convergence of gradient iterative algorithm. *J. Phys. Conf. Ser.* **2019**, *1213*, 032021. [CrossRef]
35. Hamby, D. A comparison of sensitivity analysis techniques. *Health Phys.* **1995**, *68*, 195–204. [CrossRef]
36. Statistics and Machine Learning Toolbox. Available online: https://uk.mathworks.com/products/statistics.html (accessed on 10 January 2022).

37. Film-Fit Linear Regression Model. Available online: https://uk.mathworks.com/help/stats/fitlm.html (accessed on 10 January 2022).
38. Jabr, R. Power system Huber M-estimation with equality and inequality constraints. *Electr. Power Syst. Res.* **2005**, *74*, 239–246. [CrossRef]
39. Hamby, D.M. A review of techniques for parameter sensitivity analysis of environmental models. *Environ. Monit. Assess.* **1994**, *32*, 135–154. [CrossRef]
40. van Ginkel, J.R. Standardized Regression Coefficients and Newly Proposed Estimators for R^2 in Multiply Imputed Data. *Psychometrika* **2020**, *85*, 185–205. [CrossRef]
41. Aptula, A.O.; Jeliazkova, N.G.; Schultz, T.W.; Cronin, M.T. The better predictive model: High q2 for the training set or low root mean square error of prediction for the test set? *Qsar Comb. Sci.* **2005**, *24*, 385–396. [CrossRef]
42. Gallup, J.L. Added-variable plots with confidence intervals. *Stata J.* **2019**, *19*, 598–614. [CrossRef]